

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**



# Ethernet rules closed-loop system

By John Eidson and Wesley Cole

**The enterprise tool picks up speed and moves to the plant floor.**

Traditional process control systems have used programmable logic controller (PLC)-based centralized control techniques to implement closed-loop control applications. Packet-based networks with collisions, such as Ethernet, are generally considered too slow and unreliable to safely handle closed-loop control.

Today, various fieldbus technologies and the ability to inexpensively place significant computation capability at the transducers allow control applications to be implemented using distributed techniques. This distributed function provides increased capacity, relieves computational and communication bottlenecks, and generally provides more flexibility in system design, modification, and expansion.

In a traditional process control system, the PLC polls the sensors and directs the actuators.

The PLC processor, based on the control algorithm's execution characteristics, determines the time behavior of the system. In a distributed system, the time behavior is determined by the time behaviors of the application execution characteristics on the local processors, the local protocol stacks, and the communication network. A distributed system provides true multiprocessing and therefore the potential for improving computational throughput and synchronization characteristics of the system.

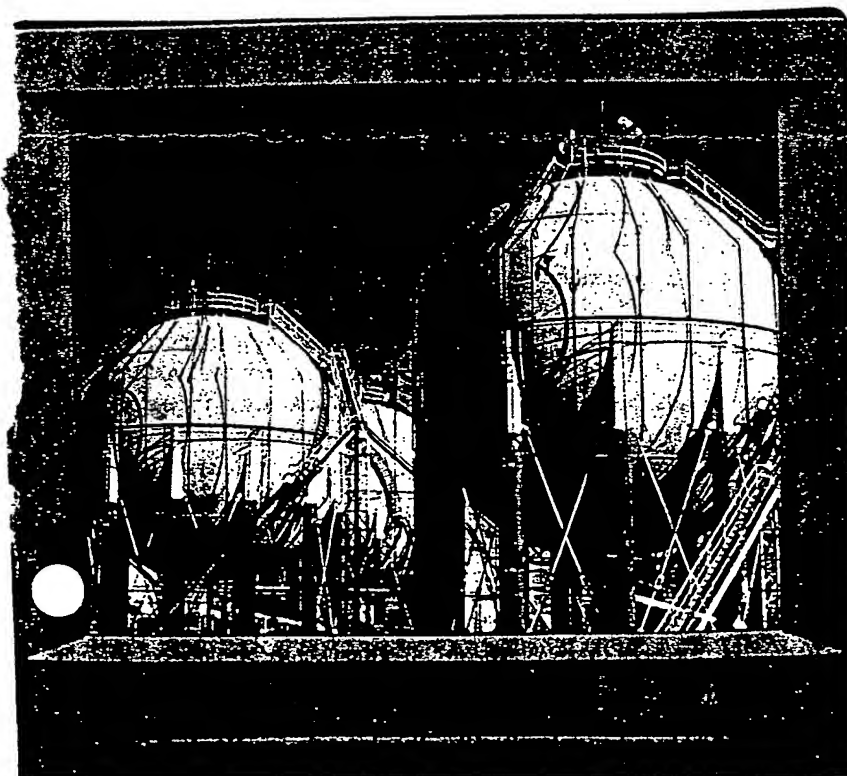
In implementing distributed systems, the process control industry has developed a wide range of communication networks. The computer industry has also provided a range of networks for the general distributed computing environment, with Ethernet being the most pervasive at the present time. Traditionally, Ethernet has not been used in control environments; it is found within the enterprise levels of process control and is increasingly being coupled to lower-level control functions. Advertisements for field-level products using Ethernet are beginning to appear in trade publications.

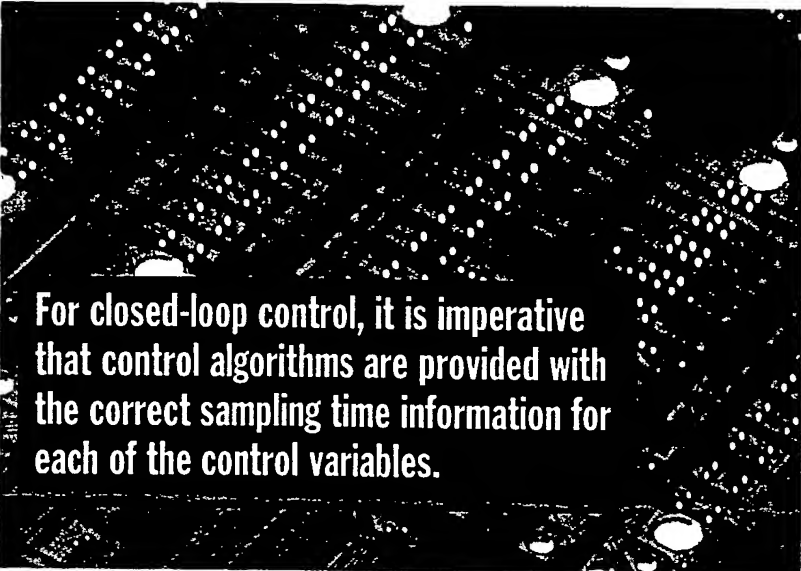
The use of general computer networks in field-level control will require some modification of techniques normally used in real-time systems. The computer industry has developed a number of distributed system techniques that will be useful in control. In particular, there is an increasing amount of literature on the use of real-time clocks in distributed systems.

Clocks are found at the PLC level of current process control systems. However, time synchronization at the device level is generally limited to timers and time ticks distributed over the network from a PLC. Let's explore the use of true real-time clocks at all levels of distributed closed-loop control systems.

## Event or data driven?

In an event-driven system, it is important that the occurrence of an event be made visible





For closed-loop control, it is imperative that control algorithms are provided with the correct sampling time information for each of the control variables.

in a reliable and timely manner. The order of events and their time relationship to the real world must be preserved. For events, time accuracy and distribution latency are the prime timing considerations. Alarms, state machines, and device commands are typical control structures using event mechanisms.

In a data-driven system, such as continuous closed-loop control, the real-world time relationship of successive data points must be preserved. Distribution latency (delay) must be such that the response times and stability requirements of the loop are not left unmet. The system throughput must be adequate to process and distribute the data.

In distributed control systems, each node must explicitly deal with synchronization issues. Messages without time stamps, when passed between nodes, can implement order, but not time specification. If time specification is to be imposed, then at least some of the nodes must have access to a clock. For closed-loop control, it is imperative that control algorithms are provided with the correct sampling time information for each of the control variables.

### Terminology

PLC	programmable logic controller
PID	proportional-integral-derivative
CSMA	carrier-sense multiple access
CAN	controller area network
NTP	network time protocol

### Provide the time

There are several ways to provide the sampling time information required by control algorithms like the widely used proportional-integral-derivative (PID) algorithm.

Polling of the sensors by the node implementing the control algorithm is commonly used today. The polling is normally periodic as seen by the polling node. The time error in polled systems is the departure from strictly periodic time, termed *time jitter*, inherent in various components of the system.

The main sources of this jitter are the communication protocol stack and the operating systems of the nodes. In the case of the protocol stack, this jitter results from queuing processes and parsing protocol headers.

Operating system time jitter arises from internal operations such as context switching and servicing software-maintained timers and clocks. With a carrier-sense multiple access (CSMA) protocol stack, such as Ethernet or LonTalk, this jitter can be important and is usually not known by the algorithm. However, even in non-CSMA stacks, such as controller area network (CAN) or Foundation fieldbus, care must be taken that operating system and application code jitter is negligible, as this jitter can be comparable to the jitter in a CSMA network.

Since there is no way to measure this jitter for each packet, there is no way to correct for jitter in simple, polled systems. A common variation on this theme is for a central node to distribute "ticks," which other nodes use to synchronize sampling. These ticks are subject to the same jitter problems as polling. In polled systems, control algorithm computations can only assume that the data was sampled at the poll times.

Another alternative is to make the entire distributed system operate synchronously. This has been done by running all the communications and computations on an enforced time-slice schedule. When run using a CSMA protocol stack, such a system can eliminate collision jitter, since by definition there is no contention for the network.

This type of setup uses synchronized clocks to enforce the schedule so data can be time stamped. Time-slice systems based on token rings or their equivalent, rather than on synchronized clocks, will have uncorrected time jitter comparable to polled systems.

The alternative presented here is to provide all nodes in the distributed system with accurately synchronized clocks. With such clocks, sensor data can be time stamped and control algorithms can use these time stamps to correct for the sample timing errors resulting from computational, operating system, and protocol stack jitter and delay. The accuracy of these clocks will determine the effectiveness of the corrections. To be effective, nodes must have hardware support for generating time stamps based on appropriate transducer control signals. Such systems can use CSMA protocols provided the clock synchronization is adequate.

As a side benefit, time stamps provide a method for detecting and correcting for missing or out-of-order data. Both missing and out-of-order data are possible in networked distributed

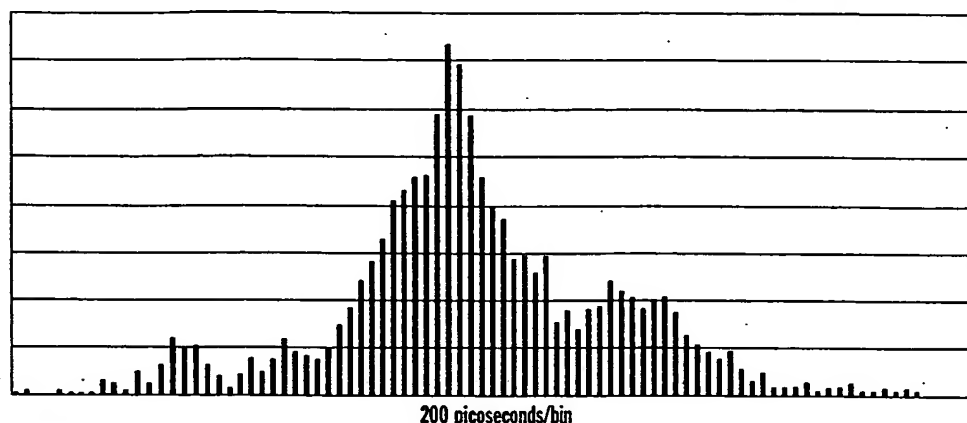


Figure 1. Time differences between two clocks using Ethernet communication

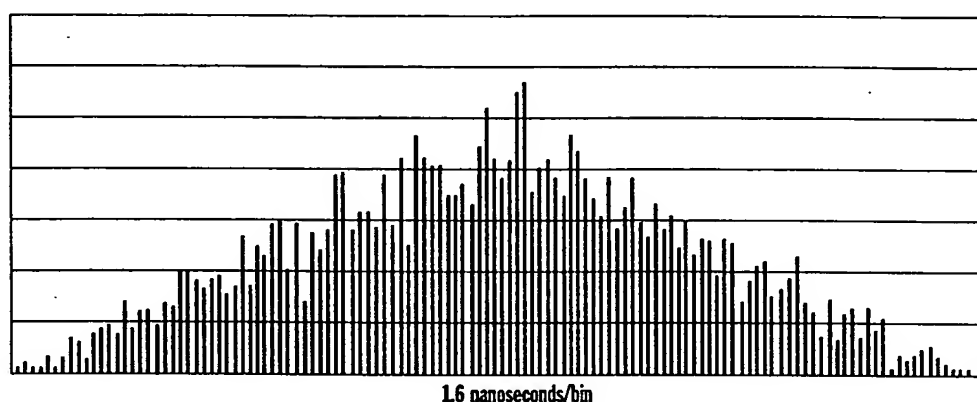


Figure 2. Time differences between two clocks using LonTalk communication

systems, especially if routers are present. However, when CSMA networks are used under the same conditions (number of nodes and network traffic) as other fieldbus networks, the occurrence of missing and out-of-order data can be made rare and can be corrected in the control algorithm.

### Synchronize your clocks

There are a number of techniques in use for synchronizing the clocks in a distributed system. A central time server is one possibility, but it introduces delays and excessive message traffic. For systems with more than a few nodes, a better choice is for each node to have a local clock participating in a synchronization protocol among the nodes.

The computer industry uses protocols such as the network time protocol (NTP) for clock synchronization among PCs and workstations. NTP is based on message exchange and produces accuracy on the order of a few milliseconds over ethernet networks. To attain better accuracy, some sort of hardware assistance is required. Systems exist that produce accuracy of 10 microseconds.

We have experimented with clock synchronization based on hardware-assisted detection and

recognition of special network packets for both Ethernet and LonTalk protocols. Synchronization accuracy is measured by recording the differences in the time stamps marking the occurrence of a series of mutually visible events as determined by the local clocks in two nodes.

Figure 1 is a histogram of these differences for two nodes communicating using the Ethernet protocol and demonstrates synchronization accuracy on the order of 20 nanoseconds. Figure 2 is a histogram of these differences for two nodes communicating using the LonTalk protocol and demonstrates synchronization accuracy on the order of 100 nanoseconds.

In both cases, this synchronization was obtained with one packet per second devoted to the synchronization algorithm. Except during the start-up phase of the algorithm, the number of synchronization packets per second is independent of the number of clocks being synchronized. The accuracy differences between the Ethernet and LonTalk implementations are due primarily to the differences in the network bit rates—10 Mbps for Ethernet and 1.2 Mbps for LonTalk.

The key benefit of this clock synchronization method is increased accuracy. While millisecond accuracy obtainable by conventional methods is adequate for many closed-loop control and monitoring applications, the submicrosecond accuracy reported here allows faster systems to be controlled in a networked environment.

#### Ethernet clocks the process

A simple control system was built to test the feasibility of implementing synchronized clocks and CSMA networks. The system is illustrated in Figure 3.

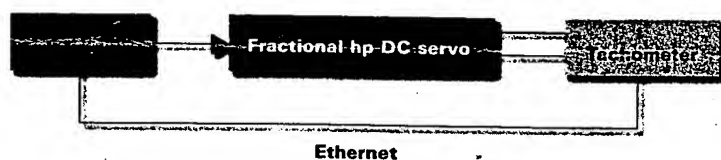


Figure 3. Test system block diagram

Both the tachometer and controller electronics were implemented using Motorola 68331 microprocessors. The communication mechanism is the unacknowledged datagram protocol on a 10base-T Ethernet. Each node contains a clock synchronized as described earlier.

The controller is a standard PID with trial coefficients determined by the Ziegler-Nichols frequency response method. The system was not stable using these initial values and was manually altered to produce a marginally stable one. Using marginal stability enabled testing under conditions that maximized the chances of observing deviations introduced by sampling jitter and network degradation.

The system was operated both in a polled mode in which the controller requested samples and in a push mode in which the tachometer controlled the sampling time. The data was always time stamped based on the local clock in the tachometer. The controller could be configured to either use or ignore the time stamps when computing its output.

As expected from previous analysis, it was difficult, but not impossible, to induce noticeable

degradation in the system. During the of the data, the packet load on the Ethernet was 25% of capacity as measured by a local-area network analyzer. In each case, 100 samples of the waveforms were collected. For each waveform, the root mean square deviation of the actual response from the mean value was computed as a percentage of the mean value. This calculation was done independently for the falling and rising portions of the square wave perturbation. The results are seen in the table below.

While more experiments are necessary to fully quantify these results, it appears that the settling time was better using the time stamp corrections.

Similar experiments were conducted in which packets were deleted to simulate network collisions or checksum errors with no visible degradation in performance. In all cases, the network packet load ranged from about 10% to 30% of capacity.

The only dramatic network-based interference produced was a result of bogus packets specifically addressed to a control system node. Such packets pass the usual Ethernet hardware filter and must be processed and rejected by the lower levels of the protocol stack, which steals processor cycles from the application. These experiments need to be repeated for controllers implementing notch filters and other algorithms that may show more sensitivity to the timing-induced phase or amplitude jitter.

Additionally, both the phase and amplitude noise introduced by time jitter should be studied, particularly that introduced by alternative algorithms and conditions. Further, evidence exists that clock usage in event-driven applications needs to be examined as well.

It's clear that the use of synchronized clocks in data-driven closed-loop control systems using Ethernet as the field bus is a viable solution. The ability to synchronize clocks to accuracy more than adequate for most control problems has been demonstrated on both the Ethernet and LonTalk protocols. The presence of these clocks allows algorithms to be corrected for the actual times of sampling, potentially eliminating computation, protocol stack, and communication time jitter. ■

#### Behind the byline

John Eidson has a Ph.D. in electrical engineering from Stanford University and has worked for Hewlett-Packard (HP) for 25 years. Wesley Cole holds a B.S.E.E. from the University of California, Davis. They work together at HP researching time synchronization and industrial automation topics.

	Percent deviation using time stamp values	Percent deviation using nominal time values
Falling half of square wave	4.69	4.73
Rising half of square wave	4.60	4.62

System response using Ethernet communication